Implementation of Multi-tier Authentication Technique for Single-Sign On access of Cloud Services

Munjpara Priyesh Pravinbhai

Department of Computer Science and Engineering
National Institute of Technology, Rourkela
Rourkela - 769 008, Odisha, India

Implementation of Multi-tier Authentication Technique for Single-Sign On access of Cloud Services

Dissertation submitted in

May 2014

to the department of

Computer Science and Engineering

of

National Institute of Technology, Rourkela

in partial fulfillment of the requirements

for the degree of

Master of Technology

by

Munjpara Priyesh Pravinbhai

(Roll No. 212CS2113)

under the supervision of

Dr. Pabitra Mohan Khilar

Department of Computer Science and Engineering

National Institute of Technology, Rourkela

Rourkela – 769 008, OdisIndia

dedicated to my parents and friends...

Department of Computer Science and Engineering
National Institute of Technology, Rourkela
Rourkela-769 008, Odisha, India.

May 2014

Certificate

This is to certify that the work in the thesis entitled ***"Implementation of Multi-tier Authentication Technique for Single-Sign On access of Cloud Services"*** by ***Munjpara Priyesh Pravinbhai***, bearing roll number **212CS2113**, is a record of an original research work carried out by him under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of *Master of Technology* in *Computer Science and Engineering*. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Dr. P. M. Khilar
Assistant Professor
CSE Department
NIT Rourkela

Acknowledgement

I am grateful to everyone who supported me throughout my thesis work. I would like to specially thank *Dr. P. M. Khilar,* Assistant Professor, Department of Computer Science and Engineering, National Institute of Technology, Rourkela, my supervisor, for his consistent encouragement, incalculable guidance and co-operation to carry out this project, and for giving me an opportunity to work on this project and providing me with a great environment to carry my work with ease.

I would like to thank all my friends and lab mates for their encouragement and understanding. They made my life beautiful and helped me every time when I was in some problem.

Most importantly, none of this would have been possible without the love and patience of my family. My family, to whom this thesis is dedicated to, has been a constant source of love, concern, support and strength all these years. I would like to express my heart-felt gratitude to them.

Munjpara Priyesh Pravinbhai

Abstract

In the last few years, Cloud computing is one of the most emerging technology in the world of computation. This technology provides services with one of the three service models: Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). This model allows users and organizations to share their vital information, crucial computing resources across the world. Securing these essential resources from the unauthorized access of the users is one of the major issues which leads to reduce the growth of this technology in the Information Technology (IT) Industries. Authentication is one of the major security parameters while providing access of the registered services to the intended users. Single-tier authentication relies on username and password for accessing the registered services which is not sufficient to secure from some well known attacks like brute-force attack, replay attacks, etc. so a solution to this issue; we come up with multi-tier authentication using single- sign on access of registered services. Multi-tier authentication security relies on the username and password as well as pattern matching and one-time password (OTP).

***Keywords*:** Cloud Computing, Authentication Techniques, Multi-tier Authentication, Single-Sign On (SSO) Access.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

In the world of computer science, during the 60s and 70s, the computation has been done by client-server architecture (Centralized Computing). This technology has been changed to distributed computing with the development of computing technologies. However, nowadays, the computing technologies again going back to the virtual, centralized computing (Cloud Computing). The cloud computing concept was first proposed by Eric Schmidt in 2006.

Cloud computing model allows access to information and computer resources using a delivery of computational services (e.g. Online file storage, social networking sites, webmail and online business applications) which allows to access software and hardware that are managed by a third party at remote locations.

The following definition of cloud computing is given by NIST: "Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g. Networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider's interaction and has been developed very quickly in the recent years [6]." This new paradigm came up with essential characteristics, service models and deployment models.

Essential Characteristics [6] [7] [8]:

Abstraction: A cloud environment hides the computational details from the users and developers. Users even don't know where their computation has been done. However, their computation should be done using registered resources only. As the abstraction level increases, underlying implementation should be less to know.

Virtualization: In computing, virtualization is the creation of a virtual (rather than actual) version of hardware, operating system, and storage device or network resources. In a cloud computing environment, virtualization can be achieved by resource pooling and resource sharing to make resources highly scalable.

On-demand self-service: A user can provision, monitor and manage the services which are automatically provided by the service provider without the help of user interaction. Computer services such as email, applications, network or server service can be provided without the help of user interaction with the service provider.

Broad network access: A user can access resources provided by a service provider using the client platform like laptops, mobile phones, PDAs, etc.

Resource Pooling: For serving computational resources to multiple users, resources are shared using a non-dedicated manner so that virtual resources dynamically assigned and reassigned according to the customer demand. Examples of computational resources are data storage, network bandwidth, virtual machines, physical memory, processing.

Rapid elasticity: Resources can be rapidly set up to quickly scale out and scale in and on an as-needed basis resources should be appearing unlimited and can be purchased in any quantity at any time to a user.

Measured service: Computing resources (e.g. User accounts, processing, storage and bandwidth) are monitored, controlled and reported transparently to both the consumer and provider by providing visibility of the rate of consumption and associated costs.

Service Models [6]:

Software as a Service (SaaS): Service providers deploy services to the web which provides remote access to the end user for accessing capabilities. The end user can utilize these services through the web interface. This model provides highest service abstractions and lowest resource visibility. This model hides the implementation of the application to the consumer. Services and underlying cloud infrastructure are not managed or controlled by the consumer. Example: gmail.com [6] [9].

Platform as a Service (PaaS): This model provides a platform to the developer to develop and deploy applications onto the cloud infrastructure by providing programming construct and tools, which can be supported by the providers. This model provides higher service abstraction than SaaS and lower resource visibility than SaaS. Deployed applications can be controlled by a consumer, but has no control over the underlying cloud infrastructure. Example: Google App Engine [6] [9].

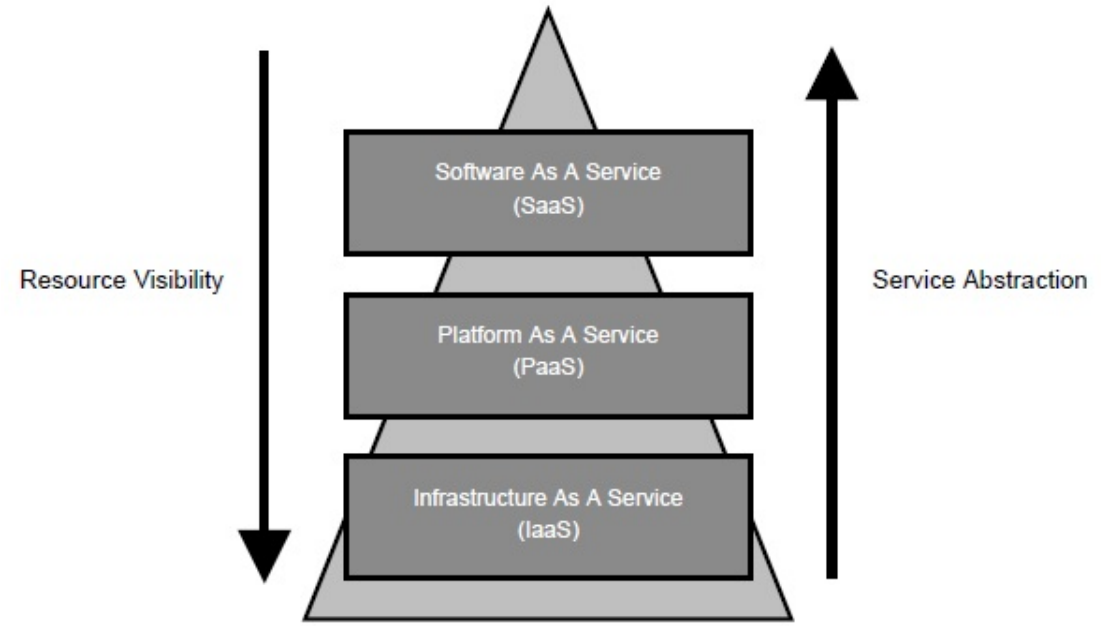

Figure 1.1: Service Oriented Architecture in Cloud Computing

Infrastructure as a Service (IaaS): The consumer is able to deploy and run any application onto the fundamental resources which is provided by IaaS providers.

This model has lowest service abstraction and highest resource visibility. The consumer has control over operating system and application, but doesn't have control over the underlying cloud infrastructure. Example: Amazon AWS [6] [9].

Deployment Models [6]:

Private cloud: The cloud infrastructure is managed or controlled by the particular organization or third party which is operated for particular an organization [9].

Community cloud: The cloud infrastructure is shared by several organizations for particular concerns like mission, security requirements, policy which can be owned or managed by third parties or the organization [6].

Public cloud: The cloud infrastructure is available to the general public with shared purpose which can be owned or managed by third parties who are providing cloud services.

Hybrid cloud: An organization can use the combination of any two or more of the above models to cloud deployment for taking advantages of individual deployment model.

Researchers mainly concentrate on hardware and software from the past few decades for improving the technologies. Due to the improvement of the technologies, the internet technologies are growing very fast across the world. So many works and services have been done online. These services include entertainment, financial transactions, gathering information about different things, emails for keeping in touch with friends, communicate with friends, whether conditions. It requires some type of authentication for all these services.

Security is one of the major issues in cloud infrastructure for adapting the cloud computing technology in IT industries. In cloud computing paradigm, the third party is providing processing capabilities, space for storing information, support for services, etc. Many organizations are storing their crucial information in the cloud database in a cloud environment. Third party maintains the cloud database. The user has to prove their identity to the service provider for seeking

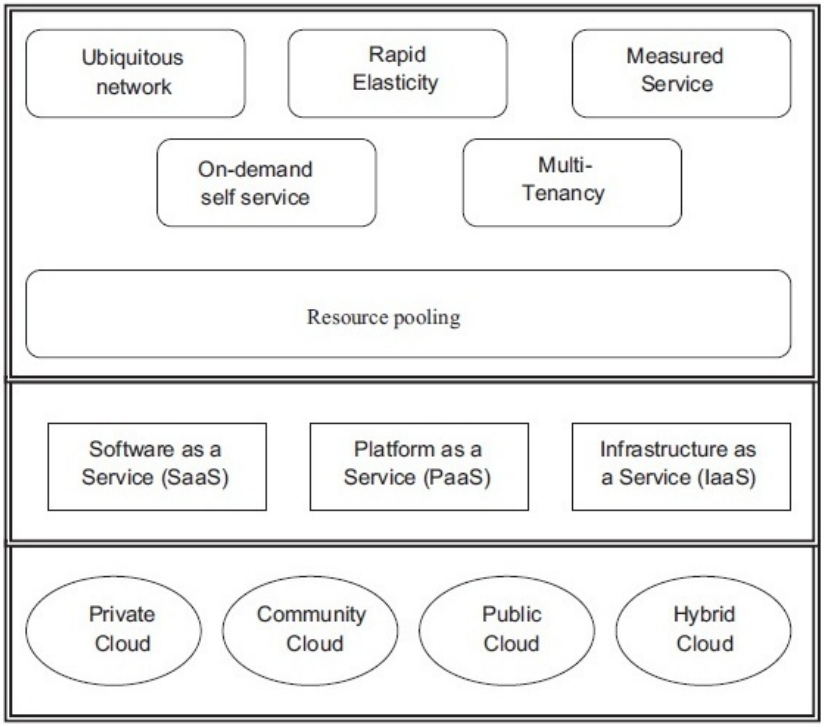

Figure 1.2: Complexity of Cloud Environment [1]

the permission of accessing requested services for utilizing the consumed resources. For taking the control over any financial transaction, the user has to pass one authentication phase for accessing the cloud service.

Authentication is the process of verifying "who you are". In case of e-mail, we require information security to carry out communication with the intended recipient. Information like different authentication parameters of the user, contacts of recipient.

The verification process has been done by one of the three types of confirmations:

Something known: Secret thing is only known to the user that can be verified by the service providers. Examples are pin no, password, private key.

Something possessed: Something that verifies the users' identity. Examples are ATM card, drivers' license, smart card.

Something inherent: Something that is inherent properties of a user. Examples are fingerprinted, retina scan, and voice.

There are three major techniques for authentication:

Password based authentication: The oldest and simplest method of authentication for accessing the resources in which user has to provide a password which is only known to the user.

Challenge-Response authentication: In this technique, users have to prove that they know the secret without sending it to the service provider. The challenge is any time stamp value which is sent by the service provider and user applies a function on challenge to send response to the service provider.

Zero-Knowledge authentication: In this technique, the user does not disclose anything that might take a chance to the confidentiality of the secret. The user proves to the service provider that they know the secret without disclosing it to the service provider. User and service provider exchanges some messages to each other for authentication. After exchanging these messages, service provider somehow knows that the user knows the secret.

Single-tier authentication can be implemented using one these techniques, but still single-tier authentication is not enough to secure the resources of the service providers in a cloud environment because this technique is suffering from many security attacks like, brute-force attack, insider attacks, in a cloud environment. For making more secure authentication model, the researcher came up with multi-tier authentication (also known as multi-factor authentication). This new technique leads to less probability of breaking the authentication system which provides more security to the resources of the cloud providers. The multi-tier authentication technique uses two or more verification process to verify the user.

In this chapter, Section 1.1 describes the background of the cloud computing technology and different authentication technique. Section 1.2 motivates us to do work with different authentication technique. Section 1.3 describes the problem area in brief. Section 1.4 shows the research purpose and goals to be achieved. Section 1.5 gives the outline of the thesis in brief. Section 1.6 gives the overall summary of this chapter.

1.2 Motivation

Nowadays cloud computing as open source technology is becoming very popular due to on-demand access of cloud services in the world of IT industries. The cloud services are distributed by the service providers on the cloud platform across the internet and the users can access these services from the simple web browser or cloud service APIs. The consumer's essential requirement is to view and manage the deployed application or consumed resources and overall control over the cloud deployment. The service providers provide the easy-to-use dashboard so that user can log in to the dashboard using service management console to manipulate the consumed resources.

For application development, security is always overhead for them in the war of functionality and security of the application. Due to the different attacks on application, an application developers' awareness about the security is linearly increasing.

One of the key challenges while developing the application is the problem of identity management in a cloud environment. Depending upon the requirement of the application, the developer implements the authentication technique which authenticates the registered user and renders the particular service for that user. This identity management scheme can easily be implemented by simple web services using single-tier authentication [9]. But single-tier authentication is not enough to secure the cloud services because, for breaking this authentication system, the hacker has to crack the password in one way which can be achieved by brute-force attack, insider attacks. The solution to this problem is multi-tier authentication. This technique has lesser probability to break the system as compared to single-tier authentication.

1.3 Problem Area

Cloud providers offer cloud services to the consumer as per their demand with different benefits of cloud computing, like reducing run time and response time,

minimize infrastructure risk, lower cost of entry, increased pace of innovation. Many security issues are coming with the cloud infrastructure which leads to impediment to the growth of cloud computing in the IT industries. There is continuous research is going on the solutions of security challenges in a cloud environment. Security as a Service is a new advancement of security solution for the cloud environment in a central unified way [10]. This approach is under the development stage to provide centralized security to cloud consumer for complete satisfaction with respect to the security of their information in the cloud. This solution should have a module for diaphanous and simplified identity management to provide the access of the cloud services to the service consumers.

This thesis project is concentrated on the authentication of a registered user in the cloud environment for providing the access of the requested service with a secure manner using multi-tier authentication technique.

1.4 Research purpose and goals

Initially, the research is to study about the cloud computing and their essential characteristics, service models and deployment models. Deciding the particular cloud platform for implementing the prototype of the thesis project is the essential step, from the different perspectives like, delivery of services, the complexity of the cloud architecture.

The main purpose of this thesis is to provide a secure authentication technique which identifies the user and delivers the requested services to the intended user in a cloud environment.

The major goals of this thesis are:

- Design a secured authentication model of cloud environment to identify a registered service consumer and renders the requested service and loads the service on the consumer's cloud interface.
- Modify the designed authentication model to single-sign on access of

registered services in which consumer has to provide their credentials once only to access a selected service.

- Implement and deploy a prototype of the designed authentication model for analyzing the security and space requirement and compare it with the existing model.

The thesis work is focused on for designing the secured authentication model with single-sign on access of registered services in a cloud environment. We used the Platform as a Service (PaaS) service model to implement and achieve this goal. Google App Engine (GAE) is a Platform as a Service (PaaS) offering by Google Co. We simulated this authentication model using GAE as a cloud platform and Google App Engine Data Store as database.

1.5 Outline of Thesis

The thesis report composed of four chapters and it is organized in the following ways. The chapter 1 presents the background of cloud computing as well as the authentication techniques, motivates for the multi-tier authentication techniques, discussing about the problem area in brief and also discussed about the research purpose and goals that are going to be achieved. The chapter 2 presents review about the existing authentication techniques and also discuss the limitations about the existing techniques. The chapter 3 discussed about the multi-tier authentication technique in the cloud along with problem associated with this technique and gives a new technique to overcome the problems with the results of the new technique which are the comparison between the existing technique and the proposed technique. The chapter 4 presents the conclusion of the thesis research and gives some future scope to extend the proposed authentication technique.

1.6 Summary

This chapter gives the introduction about the cloud computing technology and motivates to develop the authentication technique to provide access of the cloud services. It also describes the problem area with the different authentication techniques ,and research purposes and goals to be achieved in the project. Finally, it gives the outline of the thesis with the description of the each chapter in brief.

Chapter 2

Related work

This chapter introduces the single-tier and multi-tier authentication techniques and study related to its security analysis for strengthening the existing authentication techniques. It also describes the study about existing authentication techniques and its deficiencies with respect to cloud environments.

For securing confidential information of the user, authentication is a crucial security parameter. If the system is not using any authentication techniques, then confidential information can be accessed by the unauthorized persons who can use this information for making fraud in the system, or also user uses this crucial information for cheating in the financial transactions in the business perspective. Unauthorized users make the access of crucial information to crashing the system or causing the system.

In this chapter, Section 2.1 shows the reviews of the different authentication techniques. Section 2.2 gives the limitations of the existing authentication techniques with different security parameters. At last, Section 2.3 gives the overall summary of this chapter.

2.1 Review of different authentication techniques

All the application uses any authentication technique to secure the users' information in multi-tenancy way. Generally, username and password are the simple way to achieve authentication, but this technique is not that much secure to protect the information from the unauthorized access of the user. Many security attacks are possible to break this single-tier authentication system like brute-force attack, insider attack, password guessing, etc. So, [11] and [12] describes a new authentication technique that is called multi-tier authentication or multi-factor authentication. [11] describes different authentication and authorization model which states that all the application should use more than one-tier for authentication to secure the information. [11] also specifies that use one time generated secret code which will be sent to the email address or mobile number.

There are various techniques that can be used to authenticate a user. These techniques include a user's password, personal information numbers (PINs), digital certificates using public key infrastructure (PKI), and physical devices such as smart cards, one-time passwords (OTP) or other type of 'tokens', biometric identification, and other [12]. [12] includes various factors for authentication. Those are:

- **Shared secrets [13]:** A user and service providers only know the secret password (something a person knows), nobody else knows the secret. This secret makes you differ from other users for accessing the same service. An authentication system verifies the password or shared secret to access the requested service. Examples are a password, a private key.

- **Tokens [13]:** Physical devices (something the person has) are generally known as tokens. Tokens are used in all the two-factor authentications. In authentication system, first user has to provide their username and password as a first-tier authentication and then these tokens are used as second-tier authentication. Examples are a ATM card, a smart card.

- **Biometrics [5]:** Biometrics (something the person are) technique is used

to verify the user by their physical characteristics.

- **Non-Hardware base one time password (OTP) scratches card [5]:** A scratch card has been given to user for one time password. At the time of authentication, the user had to provide a secret number which is provided on the scratch card.
- **Out Of Band (OOB) authentication [5]:** This authentication technique is acting as a two-tier authentication in which user has to provide a username and password as a first-tier and provide secret code which is received by a user on their mobile number that is acting as second-tier.
- **Internet Protocol Address (IPA) Location and Geo-Location [5]:** This technique works on the geographical position in which an authentication system detects the current location of the user and assumes that they will do another transaction from the same location.

[14] represents a technique in which users have to authenticate themselves with the single-sign on (SSO) server for accessing the multiple services. SSO will handle all the subsequent authentications for different services, once the users are authenticate to the SSO server. Single-Sign On (SSO) is an access gaining process in which user has to authenticate once for issuing services from different applications. But the disadvantage of this technique is that the entire service system is compromised if SSO server's user credentials are hacked.

Dinesha et al. [2] represents an authentication technique in a cloud environment in which the authentication system verifies the user at different access level of cloud platform. This technique generates each level password and concatenates the all level passwords to gain an access of a requested service. At each level user provides password to gain access of each level. This technique uses the multi-tier authentication model to authenticate the user which gives an advantage over single-tier authentication. The problem arises in this technique is that user has to remember each level password which is very hectic for the user. The user has to reconsider all the level passwords if the user forgot the password of each level.

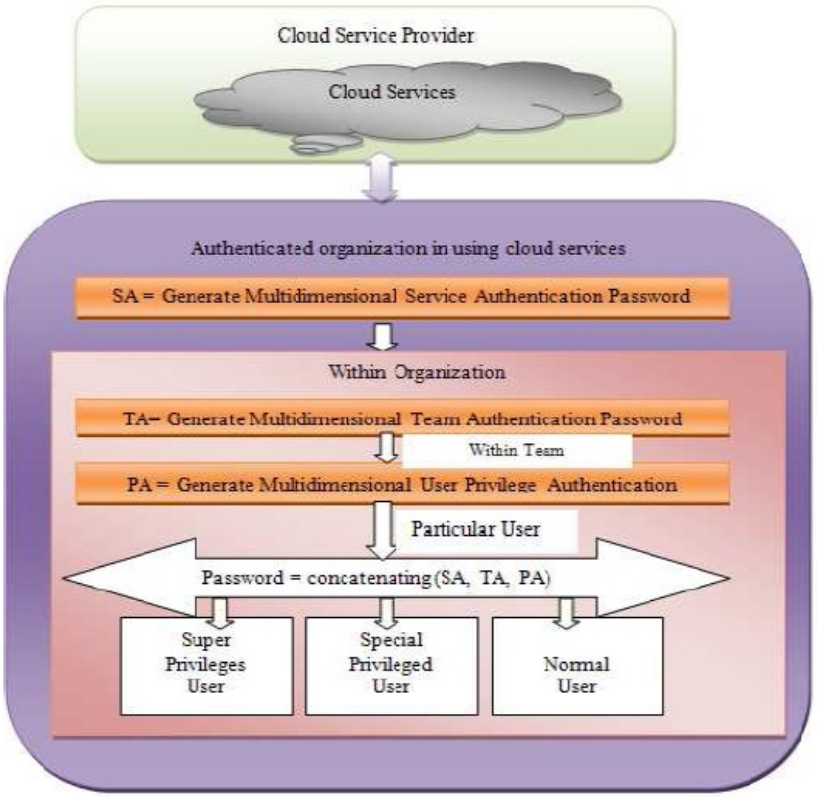

Figure 2.1: Architecture diagram of multi-level authentication system [2]

The major problem associated with cloud computing is that the user has to store their confidential data on the cloud data servers. The security of this information is the crucial aspect of the user, but in cloud computing technology, the user has to totally depend on the security policies of the cloud service providers. Prashant et al. [15] proposed one technique which is used to overcome this problem associated with cloud computing technology. This technique states that the private cloud security depends on the user. The data should be compatible with the private cloud security policies. The data should not be accepted and deny the service access if the data is not compatible with the private cloud security policies. The user can add their own security policy within the private cloud is one of the key advantages of this technique. Many disadvantages are also associated with this technique that is the initial cost of developing the private cloud and its overhead is also increased as well as the user should have to maintain this infrastructure.

Wenjun Zhang et al. [16] represent cloud architecture for Rich Internet Application (RIA). In this type of applications, the control over information is totally depends on the server side. To minimize this control of server side and maximize the

business logic towards the client side, this architecture is introduced. In this architecture, on client side, business and transaction logics, web services, and user interface are implemented by RIA whereas, on the server side, the major operations are minimized and made easy for saving and querying the data via Amazon's simple DB cloud [16]. All the major functionality is nearer to client side which leads to maximum control under the user and storing information in the database is one of the major advantages of this architecture.

Alman et al. [3] proposed a framework which provides an identity management, mutual authentication, and session key establishment between the user and the cloud server. The proposed framework authenticates the user using two-tier authentication over the cloud service access, in which first step of authentication has been done by password whereas the second step has been done with tokens and OOB authentication. The prime advantage of this framework is that it provides the maximum control under the user in terms of authentication and also protects from the attacks. The disadvantage comes in this framework is that we need extra hardware and software for carrying out the complete authentication process, which leads hectic task for the user as well as service providers.

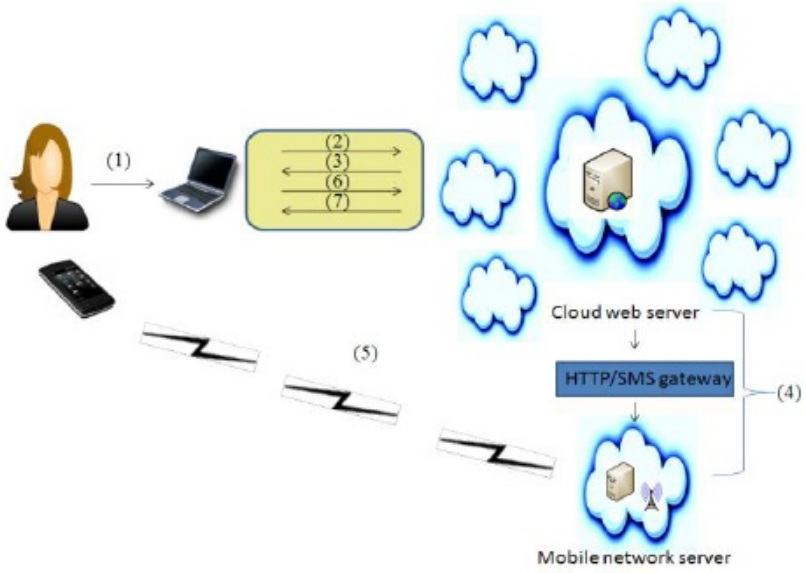

Figure 2.2: Architecture of proposed work by [3]

Fengyu Zhao, XimPeng, Wenyun Zhao et al. [17] presented security architecture for Service Oriented Architecture (SOA) applications in which these applications are

viewed as security domain and three-tier domain was divided based on the security domain analysis. Each security sub-domain has its own security requirements and develops a security model for those sub-domains.

Ahmad, Zubair, J. A. Manan, and Suziah Sulaiman, et al. [18] proposed a single-sign on (SSO) authentication model for an open environment to combine the trusted module security and platform trust in federated user systems. There is no involvement of third party in each and every transaction, such as identity or authentication service provider [18]. The identity provider or authentication service is handled by the user platform.

Deepak Bagga and Ms. Shilpi Harnal et al. [4] are proposing an authentication model for cloud computing based on the Kerberos protocol to provide single sign-on and also used to secure against DDOS attacks. Generally, for authenticating a client on the cloud computing platform, the user needs high computation and memory usage of the system. This model restricts an unauthorized access of the user and reduces the above usage of the cloud platform. This model also works as a third party between the cloud service provider and the user for providing authentication for user and secure from the DDOS attack from the attacker.

Yassin, Ali A., et al. [19] proposed a two-factor authentication technique which is based on Schnorr digital signature and feature extraction from the fingerprint to overcome the single-tier authentication issues. The major advantage of this technique is that it depends on the two factors for authentication of the user. But this technique also suffers from the additional use of hardware and software for authentication.

Nayak, Sanjeet Kumar, Subasish Mohapatra, and Banshidhar Majhi, et al. [20] proposed an authentication framework in the cloud environment. This framework specifies mutual authentication and session key agreement between the authenticated user and the cloud service provider for the completion of the authentication process. The security analysis of this technique states that it is secured from the reply attack.

[5] is proposed new two-tier authentication technique in which the authentication

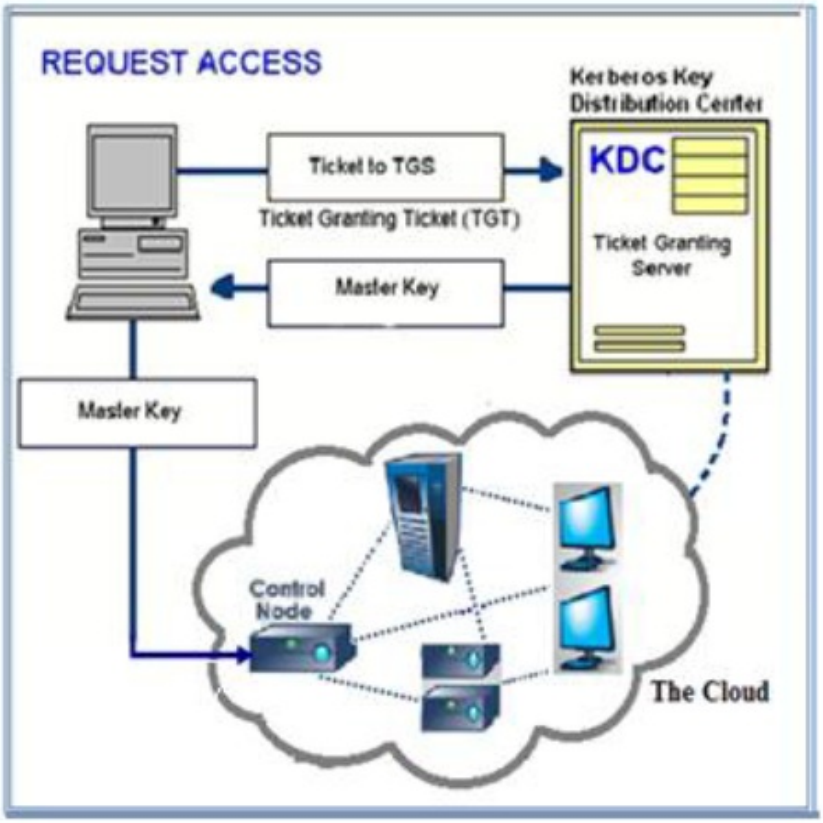

Figure 2.3: The Proposed model for accessing services using Kerberos [4]

process is carried out in two phases. After completing the two phases, user is allowed to use a requested service provided by the cloud service provider. In the first phase of authentication, the user has to provide a username and password as a normal authentication technique using some encryption, decryption techniques, and, in the second phase, the user should provide a sequence of some predetermined steps, which are submitted at the time of registration, for the authentication. To capture the second tier authentication credentials, a cloud server load a fake screen, i.e. pattern matching screen at the users' web browser. This fake screen observes the users' second-tier authentication predetermined activities for loading the requested service. If the user provides login credentials at both the phases of the authentication, then cloud server loads the original screen of the requested service. The predetermined activities in second-tier authentication of the above technique could be any of these menu activity, mouse activity, or text field activity.

[21], [22], [23] and [24] discussed different architecture models for handling privacy of information, trust between the communicating party and different access policy

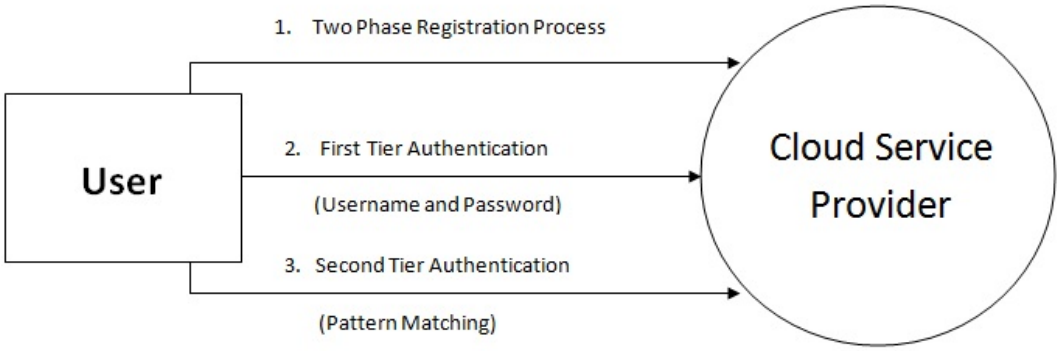

Figure 2.4: Abstract authentication model proposed by [5]

of the services but the authentication of the user is not carried out in multi-tiered way. So the scheme proposed by Singh, Maninder, and Sarbjeet Singh et al. [5] can be adopted in [21], [22], [23] and [24] for authenticating the user in the cloud environment. [25], [26] and [27] discussed various multi-tier authentication techniques. Disadvantage of all these schemes is the authentication required extra hardware and related software which is hectic task for the user.

2.2 Limitations of existing techniques

The above section discussed about different authentication techniques with the advantage and disadvantages of those reviewed techniques. The limitations can be categorized into four different security parameters.

The four security parameters are:

1. **Security from insider attack [2]:** An insider attack is a malicious attack which is happening on the network or system by gaining the authorized access to that network or system. This leads to gain access to the system by breaking the first-tier authentication. This is not acceptable by any user or organization. This leads to the use of second-tier authentication to secure the network or system.

2. **Presence of authentication control towards server or client [2]:** In cloud computing, the user or organization has to store their information in the cloud database of the cloud service provider. So we need some type of

mechanism which authenticates the client at their side. This leads to give some control over the system to the user itself [17].

3. **Extra hardware software needed [2]:** Many authentication techniques described above required some type of extra hardware and software. This adds performance degradation in the authentication techniques. This whole system depends on the extra hardware, that is, failure of the extra hardware leads to the failure of the authentication process.

4. **Number of security tiers [2]:** Single-tier authentication process is less secure than the multi-tier authentication as stated by [12] and [13] because single-tier authentication suffers from many attacks like brute-force attack, replay attacks, insider attack. So, there is a better choice to verify user using more than one-tier for accessing the service.

Table 2.1 shows the limitations of the different authentication techniques using the above discussed security parameters.

Table 2.1: Comparison of various authentication techniques [5]

Authentication Technique	**Security from insider attack**	**Presence of authentication towards**	**Extra hardware software needed**	**No of security tiers**
Authentication using Single-Sign On (SSO) [14]	No	Server	No	1
Multi-level authentication [2]	Yes	Server	No	More than 1

Continued on next page

Table 2.1 – *Continued from previous page*

Authentication Technique	Security from insider attack	Presence of authentication towards	Extra hardware software needed	No of security tiers
Architecture based on proactive model [15]	No	Server and Client	Yes	1
2-tier architecture with maximized RIA [16]	No	Server and Client	No	2
Strong user authentication framework [3]	Yes	Server	No	2
Multi-tier security feature model [17]	No	Server	No	2
SSO authentication model using Kerberos [4]	Yes	Server and Client	No	2
Anonymous Password Authentication Scheme [19]	No	Server	Yes	2

Continued on next page

Table 2.1 – *Continued from previous page*

Authentication Technique	**Security from insider attack**	**Presence of authentication towards**	**Extra hardware software needed**	**No of security tiers**
Mutual Authentication Framework [20]	No	Server and Client	Yes	2
Multi-tier Authentication Scheme [5]	Yes	Server and Client	No	2

2.3 Summary

This chapter describes the reviews about different authentication techniques proposed by different researcher and also shows the limitations associated with the proposed techniques using four security parameters.

Chapter 3

Multi-tier authentication technique using Single-Sign On access of cloud services

3.1 Introduction

In this chapter, we discussed about the multi-tier authentication technique proposed by Singh, Maninder, and Sarbjeet Singh et al. [5], in detail. It has been observed that there are some deficiency and attacks are possible with this technique. We proposed one authentication model to overcome problem associated with existing technique. By seeing security analysis and implementation results of the proposed technique, it gives better results than the existing model.

Section 3.1 gives introduction of this chapter. Section 3.2 describes the multi-tier authentication technique for cloud with associated problems. Section 3.3 describes the proposed authentication scheme to overcome the problems associated with the existing technique. Section 3.4 shows the implementation details and the results of the implementation for analyzing the security. Section 3.5 shows the comparison between the existing authentication technique and proposed authentication model with different comparison parameters. Finally, Section 3.6 gives the overall summary of this chapter.

3.2 Multi-tier authentication technique for cloud

3.2.1 Overview

This scheme uses two phases of authentication for authenticating the user on the cloud. The first phase of authentication is done by username and password, and the second phase is using the some predetermined pattern for authentication. The second phase's pattern matching or sequence of activities can be carried out on the fake screen that is loaded by the cloud service provider on the client side to do the second phase of authentication. The sequence of activities for the second phase of authentication is registered at the time of registration. If the user passes all the phases of authentication, then the only cloud server loads the requested service on the user side, otherwise the user redirected to the login page of the service.

3.2.2 Authentication Model

This scheme follows the following steps to complete the authentication process.

1. The user provides the URL of the cloud service provider in their web browser. The request is sent by the user's browser to the server of the cloud service provider. The cloud server loads the login GUI in the user's browser.

2. The user provides the username and password (first-tier authentication credentials) in the login GUI for passing through the first phase of authentication. These login credentials are passed to the server of the service provider.

3. The cloud server verifies the user for the username and password. If these credentials are correct, then the cloud server sends back the validation reply to the one application program that is observed on the client side.

4. The observer gets the validation reply from the cloud server to verify whether the further authentication has to do or not. If the validation reply is positive,

then the observer initiates the code to load the pattern matching screen which is a fake screen in the browser. The fake screen uses the database to fetch the user authentication information to verify the second phase authentication credentials.

5. Once the data has been fetched from the database, the application program loads the pattern matching screen in the browser on the client side.

6. The application program is continuously observing the web browser of the user for verifying the user at the second phase of authentication. Some sequence of activity or pattern is the second phase authentication credentials. The observer or application program is observing these activities or patterns continuously.

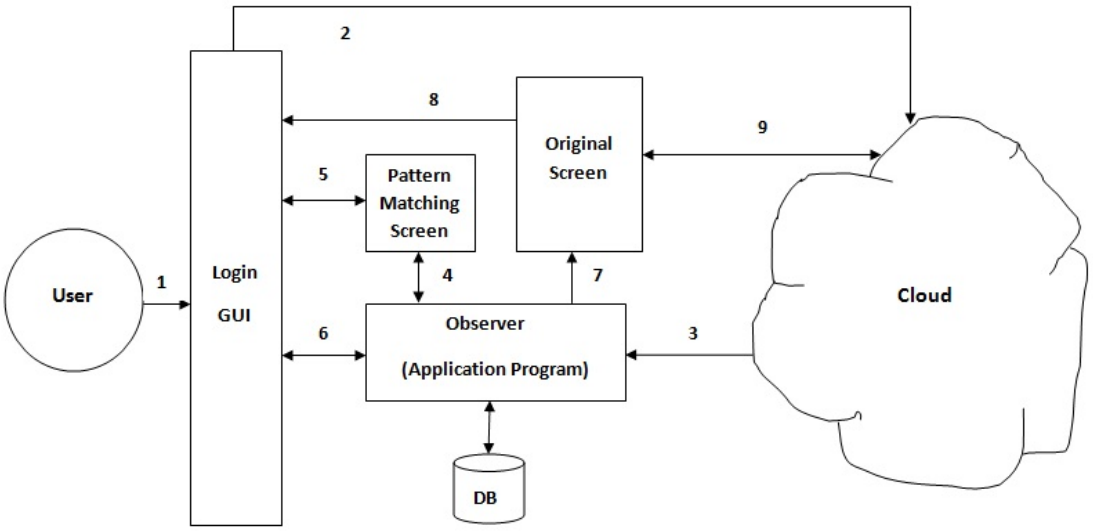

Figure 3.1: Authentication process proposed by [5]

7. If the observer observes the correct pattern of the second phase authentication entered by the user then the observer initiate the original screen of the requested service.

8. After the initiating the original screen, the observer loads the original screen in the user's web browser.

9. After loading the original screen in the browser, the cloud server makes the direct communication with the user for further activities in the loaded service.

The information fed by the user as the second-tier authentication does not depend on the external hardware or software. The sequence of activities or patterns was registered by the user at the time of registration, which will be provided by the user after he/she passes through the first-tier authentication. Singh, Maninder, and Sarbjeet Singh et al. [5] offered three techniques for the second-tier authentication, which are menu activity, mouse activity or text box activity. The service provider can use any of the techniques to implement their second-tier authentication.

1. **Menu Activity [5]:**

   The user registers the sequence of menu clicks as second-tier authentication credentials at the registration phase. This sequence has to be followed by the user after the first-phase of authentication. If the user enters the correct menu clicks at the time of second-phase authentication for authenticating himself/herself, then the observer loads the original screen of the requested service. If the user missed to follow the correct registered sequence for authentication, then the user redirected to the login page instead of the original screen of the requested service.

2. **Mouse Activity [5]:**

   The user registers the sequence of mouse clicks at a particular location on the screen as second-tier authentication credentials at the registration phase. This activity also includes the total number of mouse clicks, dragging the mouse from one position to another particular position. These different mouse events have been observed by the observer on the fake screen at the user side for authenticating the user at the second-phase of authentication. If the user follows the correct mouse events which have been registered at the registration time, then the observer loads the original screen of the requested application.

3. **Text Field Activity [5]:**

   The phrase has been registered by the user as a second-tier authentication at the time of registration. In the registered phrase, the first letter is extracted from the phrase and works as a password or pattern of the second-tier authentication credentials. Now, at the time of accessing registered service, the user has to choose the word which matches with the first letter of your

second-tier authentication password. The user has to choose a word for each and every letter of the password. Like, if the user enters the phrase "I Like To Play Cricket" at the time of registration, then the letters 'ILTPC' are extracted from the entered phase and saved as password of second phase authentication. Now, when the user asked to enter the second-tier authentication password, the user has to choose a word whose first letter is 'I', then again choose a word whose initial letter is 'L', and this will continue until the user reached for the last letter of the password. If the user chooses the correct word pattern which matches with the stored password, then the user is authenticated and the registered service is loaded into the browser of the user.

3.2.3 Problems associated with the above model

There are two major problems associated with the multi-tier authentication technique in cloud proposed by Singh, Maninder, and Sarbjeet Singh et al. [5]. It suffered from the insider attack and does not provide the single-sign on access of the registered services to the cloud service provider. In this technique, Suppose, if an insider somehow knows the authentication credentials, then he/she can access the registered services by the user without the user knowing. The single-sign on access of the service is a helpful technique for the user not to remember the many passwords to access the requested service. The single-sign on technique leads the user to access multiple services among the registered service without authenticating himself/herself every time when they want to access the services. We proposed a modified technique to overcome these two problems associated with the existing technique. A proposed authentication model design is suffering from the insider attack and single-sign on access of services.

3.3 Proposed scheme

We made some modification to the authentication technique proposed by Singh, Maninder, and Sarbjeet Singh et al. [5] to overcome the problems in the existing technique. We proposed an authentication technique by modifying the existing

two-tier authentication model to three-tier authentication with including the one extra authentication factor for verifying the intended user to overcome the insider attack and providing single-sign on access of the registered services.

The proposed authentication technique works on four phases. In the first phase, the users register themselves with the first-tier and second-tier authentication credentials. The first-tier authentication credentials are simple like username and password whereas the second-tier authentication credentials are like pattern matching or text field activity like in the existing technique [5]. We took the pattern matching as the second-tier authentication credentials to simulating the proposed scheme. For the third-tier authentication, the user does not need to provide the authentication credentials like first-tier and second-tier authentication. We are using the mobile secret code as the third-tier authentication code. This secret code is valid for some amount of time to access the requested service. We provide the time limit with the secret code. After the time limit expires, the user can not access the requested service with that secret code. The user needs another secret code for accessing the requested service. The figure 3.2 shows the abstract model of the proposed multi-tier authentication technique for single-sign on (SSO) access of cloud services.

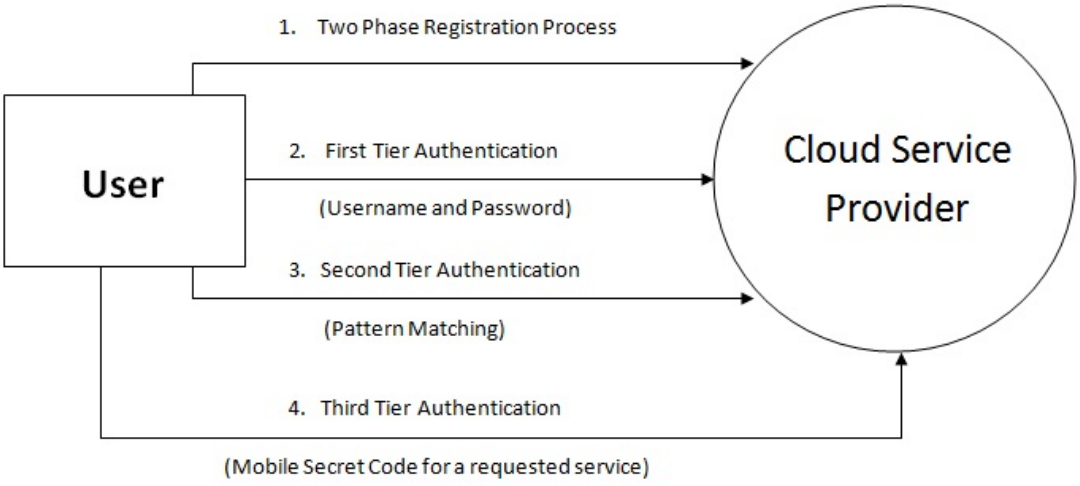

Figure 3.2: Abstract design of proposed authentication model

In the first phase, the proposed model verifies whether the intended user or not. After the first phase, the second-tier credentials are used to authorize the requested user by listing the registered services. It lists the registered services by

the user. Finally, the third-tier authentication credential is used to authenticate the requested user again and provide the access the requested service. Once the two phases of authentication is completed, the user does not need to provide the other services. For accessing the other service after the two phases, the user has to provide the mobile secret code to the authentication system.

The proposed scheme follows the following steps to authenticate the user for accessing the requested services.

1. For accessing the services, the user provides the URL of the cloud service provider in the web browser which sends the request to the cloud server for loading the Login GUI of the cloud service provider.

2. The user provides the registered username and password (first-tier authentication credentials) at the login GUI for verifying themselves to the cloud server.

3. If the username and password provided by the user to the cloud server is correct, then the cloud server sends the reply of validation at the user side. The application program or observer gets this validation reply at the user side.

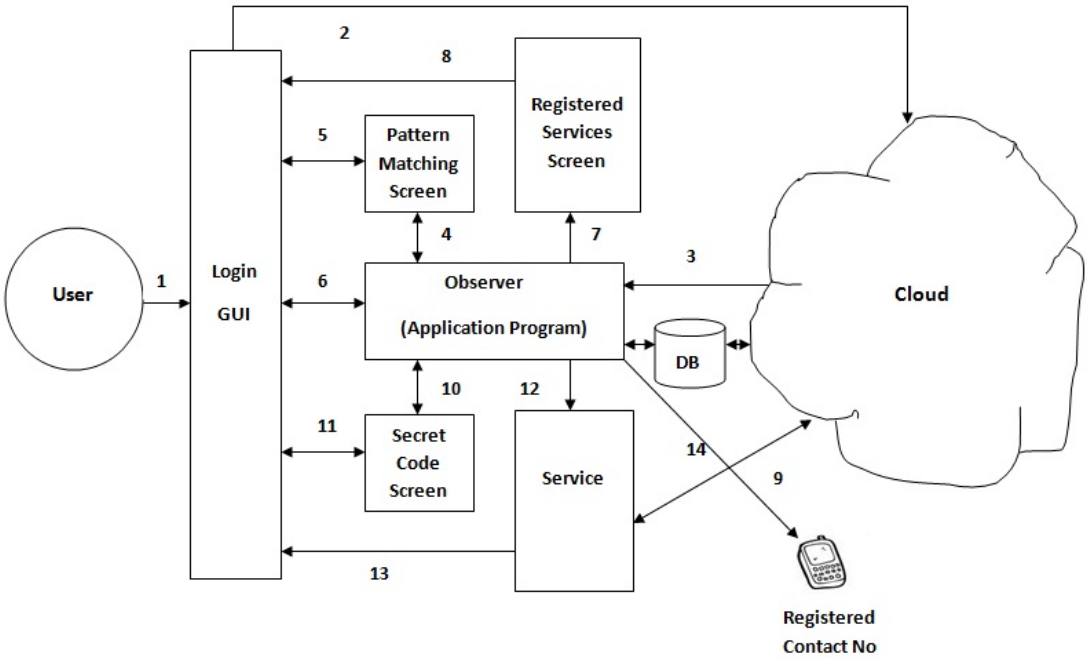

Figure 3.3: Proposed Multi-tier Authentication Model

4. The validation reply validates the user for the further authentication. If the cloud server sends the positive reply for validation, then the observer initiates the pattern matching screen for the second phase of authentication. At the same time, the observer also fetches the information from the database for the further authentication.

5. After the information are fetched from the database, the pattern matching screen (fake screen) loads into the web browser of the user.

6. The fake screen is observed by the observer continuously for authenticating the user at the second phase of authentication. The observer observes the selected pattern by the user at the pattern matching screen for further process of authentication.

7. If the correct pattern is selected by the user at the fake screen, then the application program initiates the code of the fake screen which holds the list of registered services of the user at the time of registration phase.

8. After initiating the code for the registered services screen, the observer loads the list of registered services at the user's web browser to select the particular service which is going to be used by the user.

9. Once the user select the service and submit this information to the application program, the observer sends the secret code on the registered contact number of the user. This secret code has some time limit which is set by the cloud service provider. After the time limit, the code will be expired and no more use of that code.

10. After the selection and submission of the selected service to the observer, the application program initiates the code for the secret code submission screen as well as sending the secret code the user at the same time.

11. The users provide the secret code which they got on their mobile number to the secret code submission screen for authenticating themselves.

12. Once the user provides the secret code to the observer, it will match the code which it sends to the user and also checks the time limit of that code. If the

user provides the correct secret code within the time limit, then the observer initiates the code for loading the requested service in the web browser.

13. After initiating the code of requested service, the observer loads the requested service in the user's web browser.

14. Once the requested service is loaded into the web browser of the user, direct communication has been established between the user's web browser and the cloud server.

3.4 Implementation and Results

This section describes the implementation of the extension of the multi-tier authentication technique which was first proposed by Maninder, and Sarbjeet Singh et al. [5].

3.4.1 Implementation

Our proposed multi-tier authentication technique for single-sign on access of registered services is implemented using the Google App Engine (GAE) which is provided by the Google Co. Platform as a Service (PaaS) on the cloud environment. The GAE platform is plugged-in with the Eclipse IDE for implementing the application. [28] [29] has been discussed different techniques about the development of the application on the cloud environment using the Google App Engine platform. The GAE plug-in packages [30], [31] and [32] with Eclipse IDE was used to develop the application using GAE platform on the cloud environment. RPC (Remote Procedural Call) is used to establish the secure communication between the client application and service provider's cloud server [33] and [34].

Our technique uses the Google Data Store which is provided by the Google on GAE for information manipulation by the user. The Google Data Store works as the Database System in the cloud application development. [35] discussed the basic information about the Google Data Store, and also it includes the development of the database for the application.

We have been using the Way2Sms's SMS (Short Message Service) gateway for delivering the secret code on the registered mobile number of the user. Our application uses the HTTP request and HTTP response to communicate with the SMS gateway for delivering the secret code to the intended user.

The different analysis parameters are used for analyzing the results of the implementation. These parameters are available on the Google App Engine's dashboard.

3.4.2 Results

The results of the proposed authentication model depend on the following analysis.

Security Analysis

The proposed authentication technique uses three phases of authentication. First phase used to verify using the password, second phase authorizes the user using pattern matching and finally the user authenticated with the secret code.

Let Success (S) and Failure (F) be the two outcomes of the requested cloud services.

So, the outcomes of the three authentication levels are SSS, SSF, SFS, SFF, FSS, FSF, FFS, FFF and N (O) = 8 for our proposed authentication model, where, O = outcomes.

Now, let, the p = probability of the success for accessing the services at each authentication level
So, success, SSS, for breaking the whole authentication system, i.e. multi-tier authentication system is denoted by P (E). Where, P (E) = p^3. This leads the failure for breaking the authentication system is 1 - P (E) = 1 - p^3.

Now, let say p = 0.2, then p^2 = 0.04 and p^3 = 0.008. It means the probability of success in breaking the whole authentication system is very less, almost zero, compared to one-tier and two-tier authentication system.

The strength of all the three tiers of the authentication system depends on the

password and pattern chosen by the user at the registration time and a secret code generated by the cloud server.

The strength of the authentication system is indirectly proportional to the probability of success in breaking the multi-tier authentication system. It means the higher the strength, the lesser the probability of success for breaking the system.

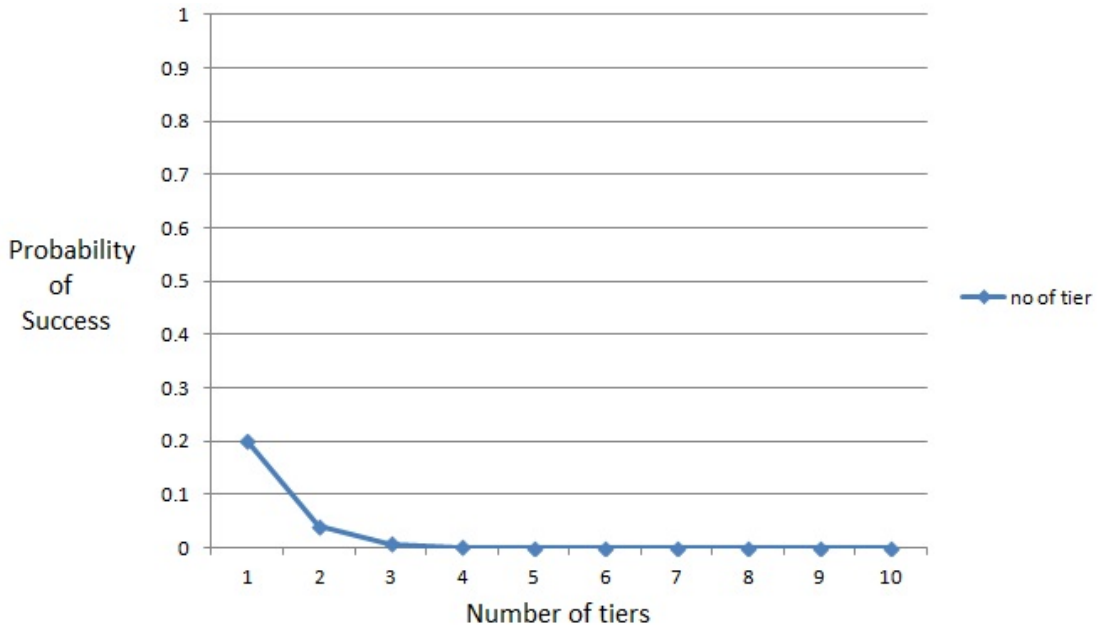

Figure 3.4: Probability of success for breaking the multi-tier authentication system

The figure 3.4 shows the relation between the probability of success and the number of tiers of the multi-tier authentication system.

Figure 3.4 clearly shows that the probability of success in breaking the multi-tier authentication system is exponentially followed with the number of tiers in the authentication system.

Space Requirements

We use the space as a second analysis parameter for our proposed authentication technique. For evaluating this parameter, we find the result of the space usage of the one-tier and two-tier authentication and analyze those results and we conclude them in the following figure 3.5. The following figure 3.5 shows the

linear relationship between the spaces required to store the user's login credentials of one-tier, two-tier and three-tier authentication system.

Figure 3.5: Memory space used by the registered users

From the figure 3.5, the memory space required by the user is linearly increasing as the number of users are registered themselves in the cloud application.

From the dashboard of the Google App Engine (GAE), we find that the space is consumed by the one user's credential for one-tier authentication, two-tier authentication and three-tier authentication system.

Application required space of one user's credentials for one-tier authentication is 217 bytes, for two-tier authentication is 625 bytes and three-tier authentication (proposed technique) is 709 bytes.

If 1 million users are registered at any moment of time, then the cloud server needs 1000000 * (709-625) = 84000000 = 84MB of extra memory space to store the users' credentials. This is not a big issue while we are comparing with the security of the data.

3.5 Comparison between existing authentication model and proposed authentication model

The following table 3.1 shows the comparison between existing authentication technique and proposed authentication technique with three comparison parameters.

Table 3.1: Comparison between existing technique and proposed technique

Comparison Parameters	**Multi-tier authentication Technique**	**Multi-tier authentication Technique (Our Scheme)**
Probability of success (p) for breaking the authentication system (let, p = 0.1)	0.01	0.001
Single-sign on access of cloud services	No	Yes
No of authentication factor	One	Two

3.6 Summary

This chapter describes the multi-tier authentication technique proposed by Singh, Maninder, and Sarbjeet Singh and shows the problem associated with this technique. It also shows the proposed authentication technique by us with implementation detail and results with comparison between the existing technique and proposed technique.

Chapter 4

Conclusion and Future Scope

This chapter concludes the overall examination about this thesis and recommends some of the future works in the research area of authentication system. Section 4.1 gives the conclusion of proposed authentication technique. Section 4.2 shows the future scope for extending the proposed technique.

4.1 Conclusion

Any authentication system's core strength depends upon the probability of success for breaking that system for accessing the services provided by the cloud service providers. In our proposed authentication scheme, the core strength is first-tier, second-tier and third-tier authentication user credentials. For getting the access of the requested service, the attacker has to break all the authentication layers.

Security analysis says that increases as the number of authentication tiers in the system, the probability of success for breaking the multi-tier authentication system reaches near to the zero. Hence, by seeing the analysis of security, we can say that there is a very less probability of breaking the multi-tier authentication system. If we consider the usability of the storage space, then the proposed technique takes more space than the existing authentication technique which is very less and also we can say that it is negligible in the case of cloud environment where large amount of storage and its scalable.

Space requirement says that the as increases the number of registered users in the cloud application, the storage space consumed by the user's credentials are linearly increases and this will not cause more processing and fetching overhead to the cloud server. For handling the pressurized situations, this technique adds the fake screen concepts. This fake screen is not related to any software and hardware.

By using the secret code on mobile mechanism, the proposed authentication technique provides the single-sign on access of the cloud services provided by the service providers. The user has to provide a secret code which is getting on the registered mobile number for accessing the particular requested service. This mechanism leads the proposed technique is free from the masquerade attack.

4.2 Future Scope

The main focus of this thesis is on the re-design and implementation of the multi-tier authentication proposed by [5] in which there is no use of external software or hardware for authenticating the user. If the user wants to change his/her password or pattern, then there is no particular mechanism in the proposed scheme. There has to be a mechanism which provides the modification of the password or pattern. The multi-tier mechanism is required for the modification of the password or pattern and also for recovering the username or password which is a future scope of this project. This proposed scheme also not provides the security of confidential information about the user which is stored on the cloud database. There are many encryption-decryption techniques and also hashing technique for securing the confidential information.

Bibliography

[1] S Subashini and V Kavitha. A survey on security issues in service delivery models of cloud computing. *Journal of Network and Computer Applications*, 34(1):1–11, 2011.

[2] HA Dinesha and VK Agrawal. Multi-level authentication technique for accessing cloud services. In *Computing, Communication and Applications (ICCCA), 2012 International Conference on*, pages 1–4. IEEE, 2012.

[3] Amlan Jyoti Choudhury, Pardeep Kumar, Mangal Sain, Hyotaek Lim, and Hoon Jae-Lee. A strong user authentication framework for cloud computing. In *Services Computing Conference (APSCC), 2011 IEEE Asia-Pacific*, pages 110–115. IEEE, 2011.

[4] Ms. Shilpi Harnal Deepak Bagga. Single sign-on authentication model for cloud computing using kerberos. 2013.

[5] Maninder Singh and Sarbjeet Singh. Design and implementation of multi-tier authentication scheme in cloud. *International Journal of Computer Science Issues (IJCSI)*, 9(5), 2012.

[6] Peter Mell and Tim Grance. The nist definition of cloud computing. *National Institute of Standards and Technology*, 53(6):50, 2009.

[7] Panagiotis Kalagiakos and Panagiotis Karampelas. Cloud computing learning. In *Application of Information and Communication Technologies (AICT), 2011 5th International Conference on*, pages 1–4. IEEE, 2011.

[8] Barrie Sosinsky. *Cloud computing bible*, volume 762. John Wiley & Sons, 2010.

[9] Rasib Hassan Khan, Jukka Ylitalo, and Abu Shohel Ahmed. Openid authentication as a service in openstack. In *Information Assurance and Security (IAS), 2011 7th International Conference on*, pages 372–377. IEEE, 2011.

[10] Davit Hakobyan. Authentication and authorization systems in cloud environments. 2012.

[11] David Chou. Strong user authentication on the web. `http://msdn.microsoft.com/en-us/library/cc838351.aspx`, August 2008.

[12] Federal Financial Institutions Examination Council. Authentication in an internet banking environment. 2011.

[13] William E Burr, Donna F Dodson, and William T Polk. *Electronic authentication guideline*. Citeseer, 2004.

[14] Ashish G Revar and Madhuri D Bhavsar. Securing user authentication using single sign-on in cloud computing. In *Engineering (NUiCONE), 2011 Nirma University International Conference on*, pages 1–4. IEEE, 2011.

[15] Prashant Srivastava, Satyam Singh, Ashwin Alfred Pinto, Shvetank Verma, Vijay K Chaurasiya, and Rahul Gupta. An architecture based on proactive model for security in cloud computing. In *Recent Trends in Information Technology (ICRTIT), 2011 International Conference on*, pages 661–666. IEEE, 2011.

[16] Wenjun Zhang. 2-tier cloud architecture with maximized ria and simpledb via minimized rest. In *Computer Engineering and Technology (ICCET), 2010 2nd International Conference on*, volume 6, pages V6–52. IEEE, 2010.

[17] Fengyu Zhao, Xin Peng, and Wenyun Zhao. Multi-tier security feature modeling for service-oriented application integration. In *Computer and Information Science, 2009. ICIS 2009. Eighth IEEE/ACIS International Conference on*, pages 1178–1183. IEEE, 2009.

[18] Zubair Ahmad, JA Manan, and Suziah Sulaiman. Trusted computing based open environment user authentication model. In *Advanced Computer Theory*

and Engineering (ICACTE), 2010 3rd International Conference on, volume 6, pages V6–487. IEEE, 2010.

[19] Ali A Yassin, Hai Jin, Ayad Ibrahim, and Deqing Zou. Anonymous password authentication scheme by using digital signature and fingerprint in cloud computing. In *Cloud and Green Computing (CGC), 2012 Second International Conference on*, pages 282–289. IEEE, 2012.

[20] Sanjeet Kumar Nayak, Subasish Mohapatra, and Banshidhar Majhi. An improved mutual authentication framework for cloud computing. *International Journal of Computer Applications*, 52, 2012.

[21] Sarbjeet Singh and Seema Bawa. A privacy policy framework for grid and web services. *Information Technology Journal*, 6(6), 2007.

[22] Sarbjeet Singh and Dolly Sharma. An access control framework for grid environment.

[23] Sarbjeet Singh. Trust based authorization framework for grid services. *Journal of Emerging Trends in Computing and Information Sciences*, 2(3):136–144, 2011.

[24] Sarbjeet Singh and Seema Bawa. A privacy, trust and policy based authorization framework for services in distributed environments. *International Journal of Computer Science*, 2(2), 2007.

[25] Charles Miller. Password recovery. `http://fishbowl.pastiche.org/archives/docs/PasswordRecovery.pdf`.

[26] Google account recovery. `https://accounts.google.com/RecoverAccount`.

[27] Peter Mell and Timothy Grance. The nist definition of cloud computing (draft). *NIST special publication*, 800(145):7, 2011.

[28] Daniel Guermeur and Amy Unruh. *Google App Engine Java and GWT Application Development*. Packt Publishing Ltd, 2010.

[29] Google web toolkit get started. `https://developers.google.com/webtoolkit/doc/latest/FAQ_GettingStarted`.

[30] Google plugin for eclipse 3.7 installation instructions. https://developers.google.com/eclipse/docs/install-eclipse-3.7.

[31] Google web toolkit: Organize projects. https://developers.google.com/webtoolkit/doc/latest/DevGuideOrganizingProjects.

[32] Create a gwt project. https://developers.google.com/webtoolkit/doc/latest/tutorial/create.

[33] Communicating with server. https://developers.google.com/webtoolkit/doc/latest/tutorial/clientserver.

[34] Making remote procedure calls. https://developers.google.com/webtoolkit/doc/latest/tutorial/RPC.

[35] Java datastore api. https://developers.google.com/appengine/docs/java/datastore/.